Efficient Security Based Information Sharing System

Badiginchala Manohari,D.Kumar
Dept of CSE
Dr.Samuel George Institute of Engineering & Technology, Markapur, India

Abstract

With the development of big data and cloud computing, more and more enterprises prefer to store their data in cloud and share the data among their authorized employees efficiently and securely. So far, many different data sharing schemes in different fields have been proposed. However, sharing sensitive data in cloud still faces some challenges such as achieving data privacy and lightweight operations at resource constrained mobile terminals. Furthermore, most data sharing schemes have no integrity verification mechanism, which would result in wrong computation results for users. To solve the problems, we propose an efficient and secure data sharing scheme for mobile devices in cloud computing. Firstly, the scheme guarantees security and authorized access of shared sensitive data. Secondly, the scheme realizes efficient integrity verification before users share the data to avoid incorrect computation. Finally, the scheme achieves lightweight operations of mobile terminals on both data owner and data requester sides.

With the rapid development of information technology and Internet of Things (IoT), enterprises generate more and more big data, which needs to be stored and processed efficiently and securely. Cloud computing is a developed storage platform and has many advantages including low cost and scalability [1,2,3]. Therefore, many enterprises and individuals are apt to outsource their data to cloud for storage and sharing with authorized data requesters. For example, in a cloud based health information system, patients upload their health information to cloud for sharing with medical experts to diagnose diseases. Similarly, the manager of an enterprise not only want to store the big data in cloud, but also want to share the data among their authorized employees wherever needed. Outsourcing data for sharing in cloud not only saves local storage space, but also greatly reduces the cost of enterprises in software purchase and hardware maintenance [4,5,6]. Although people take advantages of this new technology and service, their concerns about data security arises as well. Security problem in cloud is the most critical issue because of the valuable information that data owners share. Cloud providers should address privacy and security issues as a matter of high and urgent priority [7,8,9]. One of the prominent security concerns in data sharing is data privacy. In addition, terminals of users are usually resource-constrained mobile devices with small storage space and low processing speed. Therefore, it is essential to propose an efficient and secure data sharing scheme for mobile devices in cloud computing.

Main contributions

We propose a lightweight and secure sensitive data sharing scheme for mobile devices in cloud computing. The main contributions of the paper are as follows.

1) We design an efficient integrity mechanism based on algebraic signature for sensitive data before data sharing.
2) We guarantee privacy preserving of sensitive data in sharing process

and access authorization control for data requesters.

3) We achieve lightweight computation operations on data owner and data requester sides with mobile devices.

Organization

The rest of paper is organized as follows. Section II introduces the related works in secure data sharing. Section III describes architecture and security requirements. Section IV presents the definitions and preliminaries. In section V, we describe the implementations of the efficient data sharing scheme. We analyze security in section VI and evaluate the scheme performance in Section VII. Finally, we conclude this paper in Section VIII.

The related work

With more and more sensitive information sharing among enterprise employees, preserving data integrity and guaranteeing access control for only authorized users to access the data has become the core security problem. Therefore, security problems of data sharing in cloud mainly focus on access control and data integrity [10,11,12,13,14].At present, data sharing schemes mainly employ access control mechanism to achieve authorized access. Akl and Taylor [15] proposed use of cryptography to realize access control in hierarchical structures. As data owner and the data center are not in the same trusted domain in cloud storage system, access control schemes employing Attributed Based Encryption (ABE) [16] are put forward. ABE commonly comes in Key Policy ABE (KP-ABE) and Cipher text Policy ABE (CP-ABE). KP-ABE uses attributes to describe the encryption data and builds policies into user's key [17]. CP-ABE uses attributes to describe user's credentials and the user encrypting the data determines a policy on who can decrypt the data [18]. Lewko [19] proposed the first unbounded KP-ABE scheme. Waters [20] firstly put forward a fully expressed CP-ABE scheme in the standard model. Cheung and Newport [21] proposed another CP-ABE scheme and proved its security in the standard model. To ensure data security in smart health, Zhang and Zhen [22] realize the fine-grained access control, cipher text-policy attribute based encryption (CP-ABE). To achieve privacy and authorized access, many other schemes [23,24,25,26,27,28,29,30,31,32,33] are proposed in diverse fields and applications.

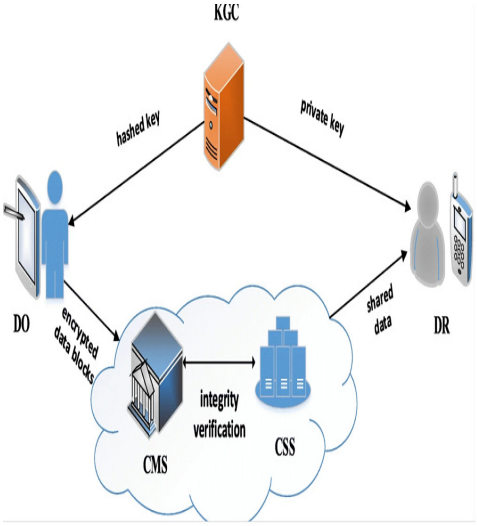
To verify integrity of data in cloud, many integrity verification schemes [26, 34,35,36,37,38,39,40,41,42] have been proposed in recent years. Ateniese [34] proposed the first public auditing scheme, which allows any public verifier to check the data integrity. Later, to prove the integrity of dynamic data, Ateniese [26] proposed another scheme based on the symmetric key provable data possession (PDP) scheme. To support dynamic operation of data, Erway et al. [35] proposed a dynamic provable data possession (DPDP) scheme by introducing an authenticated skip list. Later many auditing schemes are proposed by using the authenticated data structure to support dynamic data update. Zhu et al. [36] introduced an index-hash table (IHT) for dynamic verification. Yang et al. [37] proposed another authenticated data structure called index table (ITable) to store the abstract information of blocks. Tian [38] proposed a data structure named Dynamic-Hash-Table (DHT). Wang et al. [39] and Liu et al. [40] respectively proposed dynamic public auditing schemes based on Merkle Hash Tree (MHT). The two schemes can achieve both public verification and dynamic data operations. However, block signatures are generated by users, which would incur multitude computation and communication overhead on the user side. In 2018, Gan [41] proposed an auditing scheme on algebraic

signatures that can achieve lower computation and communication costs. The properties of algebraic signature allows cloud to return a sum of the selected blocks in the proof instead of the original data file, which saves the bandwidth between the cloud and the verifier and make the algebraic signature suited for cloud computing [42]. Nowadays, more and more schemes [23, 43,44,45,46,47,48,49,50,51,52] on cloud computing security are put forward to achieve great advantages.

Architecture and security requirement

System model

The system model consists of four entities named Key Generator Center (*KGC*), Data Owner (*DO*), Cloud Servers (*CS*) and Data Requester (*DR*) as depicted in Fig. 1.



Key generation center (KGC)

It is responsible for generating public parameters and master key for the system and issuing private key for other entities.

Data owner (DO)

It is responsible to generate and encrypt the shared data, define access structures, and divide encrypted data into blocks.

Cloud servers (CS)

Cloud servers comes in Cloud Storage Servers (*CSS*) and Cloud Manage Servers (*CMS*) based on their roles. CSS is responsible for storing shared data, block tags and supply the data integrity proof. To save computation and communication costs of mobile terminals of *DO* and *DR*, *CMS* is employed to manipulate complex computations including generating algebraic signatures of blocks, verifying data integrity of shared data and computing the intermediate data for encryption and decryption.

Data Requester (DR): It is responsible to download and decrypt the shared data for utilization. In the scheme, only the authorized *DR* is able to download shared data from *CSS* and decrypt the data.

In our secure data sharing scheme for mobile terminal devices, DO has large sensitive data to share with legitimate DR. Before sharing, DO encrypts the data with his private key and outsources the data to CSS. If a DR wants to access the data, he must register his identity to KGC and obtain his private key for decryption. To achieve authorized access, only legitimate DRs with correct attributes can download and utilize the shared data. To ensure cloud data intact and decrease computation burden of requesters, CMS helps DR to verify the integrity of data before sharing. Only when data is undamaged, DR downloads and decrypts shared data with his private key.

Security requirement

In the scheme, we suppose *CSS* and *CMS* are both semi-trusted. *CSS* is responsible to store data and block tags for data sharing. However, once data is corrupt or lost, it might launch forge attack or replace attack for economic reasons. Similarly, *CMS* is curious about

the content of sensitive data, so the data should preserve secret to *CMS*. In the scheme, we assume *KGC* is a fully trusted authority and can honestly generate private key for the system and other entity. Therefore, the following security requirements of the scheme should be satisfied.

Data confidentiality

The shared data must keep confidential to *CSS*, *CMS* and any unauthorized *DR*s for privacy and security. Any disclosure of shared data is undoubtedly harmful to enterprise benefits. Consequently, it is important to ensure the confidentiality of shared data.

Data integrity

The data should keep intact before shared by DR. It means that the data is undamaged in an unauthorized manner during storage and sharing process.

Authorized access

To achieve authorization, only *DR* with correct attributes can access shared data stored in *CSS*.

User revocation

The membership of *DR* must be revoked to stop his access to shared data when he leaves the organization. To achieve security of the scheme, user revocation should be required in the data sharing scheme.

Design goals

The data sharing scheme for mobile devices is designed to achieve data privacy preservation, data security and lightweight operations.

Privacy preservation

The scheme should satisfy data privacy during data sharing process. As sensitive data is encrypted by data owner before outsourcing to cloud and only authorized data requesters can access the encrypted data, the shared data is private to CSS, CMS and any unauthorized DRs.

Data security

The scheme should achieve sensitive data security during the whole sharing process. The security requirement can be guaranteed by data confidentiality, data integrity, authorized access and user revocation in the scheme.

Lightweight operations

The scheme should decrease computation operations of *DO* and *DR* for efficiency. In our scheme, *CMS* is responsible to divide encrypted data into blocks and computes block tags. Furtherly, when *DR* wants to access shared data, *CMS* compute intermediate data of decryption to less *DR*'s computation burden.

Definitions and preliminaries

Denifitions

1. Discrete Logarithm (DL) Assumption. Suppose g is a generator of multiplicative cyclic group GG with prime order q. On input y∈Gy∈G, there does not exist probabilistic polynomial time algorithm that outputs a value x∈Z∗qx∈Zq∗ such that $g^x = y$ with non-negligible probability.
2. Computational Diffie-Hellman (CDH) Assumption. Suppose g is a generator of multiplicative cyclic group GG with prime order q. On input gx,gy∈Ggx,gy∈G, there does

not exist probabilistic polynomial time algorithm that outputs gxy∈Ggxy∈G with non-negligible probability.

3. Access Structure. Suppose $P = \{P_1, P_2, \ldots, P_n\}$ is a set of parties. A collection of $W \subseteq 2^P$ is monotone if $\forall B,C : B \in W$ and $B \subseteq C$ then $C \in W$. An access structure is the collection W with non-empty subsets of P, i.e., $W \subseteq 2^P \backslash \{\emptyset\}$. The sets in W are named as authorized sets, and the sets not in W are named as the unauthorized sets.

Preliminaries

1. Linear secret-sharing schemes (LSSS). LSSS is a share-generating matrix A with rows labeled by attributes. Assume $S \in A$ is an authorized set and I is defined as $I = \{i | \rho(i) \in S\}$. Then there exists constants $\omega_i \in Z_q$ satisfying $\sum_{i \in I} \omega_i \lambda_i = s$ where λ_i is valid share of secret share s. suppose A_i is the i^{th} row of A, the equation $\sum_{i \in I} \omega_i A_i = (1, 0, \ldots, 0)$ also satisfies.
2. Algebraic signature. The algebraic signature of a file block composed of strings $s_0, s_1, \ldots, s_{n-1}$ is defined as sigg(s0,s1,…,sn−1)=∑n−1i=0si·gisigg(s0,s1,…,sn−1)=∑i=0n−1si·gi . The algebraic signature has the following two properties: i) the algebraic signature of a combination of file F_1 and file F_2 is defined as $sig_g(F_1 \oplus F_2) = sig_g(F_1) \oplus sig_g(F_2)$. Ii) the algebraic signature of a combination of n blocks in file F is equal to the combination of algebraic signatures of each block named mi∈Gmi∈G, which is described as ∑ni=1sigg(mi)=sigg∑ni=1mi∑i=1nsigg(mi)=sigg∑i=1nmi.
3. XOR-homomorphic function. A XOR-homomorphic function h is a pseudo-random function that can ensure data privacy. Its properties is as follows. For any inputs $x,y,$ there exists $h(x \oplus y) = h(x) \oplus h(y)$.
4. Bilinear maps. Suppose G1,G2G1,G2 are two multiplicative groups with same large prime order q, and g is a generator of G1G1. A bilinear map e is a map function e:G1×G2→G1e:G1×G2→G1 with the following properties: i) computability. ∀u,v∈G1∀u,v∈G1, an efficient algorithm exists to compute $e(u, v)$. Ii) Binearity. $\forall a,b \in Z_q, \exists e(u^a, v^b) = e(u, v)^{ab}$. Iii) nondegeneracy. $e[g, g] \neq 1$. Iv) security. It is hard to compute discrete logarithm (DL) in G1G1.

Scheme implementations

In this section, we present the efficient and secure data sharing scheme for mobile devices in cloud computing in detail. We divide the sharing scheme into four phases named initial phase, data processing phase, integrity verification phase and data sharing phase.

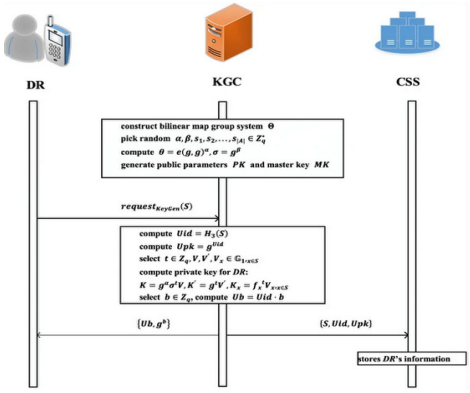
Initial phase

This phase consists of three algorithms named ***ParaSetup,KeyGen,IdReg***. Algorithm ***ParaSetup*** is mainly responsible to generate system parameters before data sharing. Algorithm ***KeyGen*** is mainly used to obtain the private key for DR to decrypt the cipher-text of shared data. Algorithm ***IdReg*** is responsible for registering DR's identity information in a table for checking the validity of DR. Figure 2 describes the data flow of the phase.

1. *ParaSetup*(λ, A) → (PK, MK):It is run by *KGC*. Given system security parameter λ, *KGC* constructs the bilinear map group

system Θ=(G1,G2,q,e)Θ=(G1,G2,q ,e) where G1,G2G1,G2 are multiplicative groups with prime order q, and e is a bilinear map e:G1×G1→G2e:G1×G1→G2. Suppose A is the attribute universe A whose attribute number is $|A|$. *KGC* picks random α,β,s1,s2,…,s|A|∈Z∗qα,β,s 1,s2,…,s|A|∈Zq∗ and computes $\theta = e(g, g)^{\alpha}$, $\sigma = g^{\beta}$. Then KGC defines heteromorphic function h :G1→G1:G1→G1 and algebraic signature $sig_g(m_i) = m_i \cdot g^i$, where mi∈G1mi∈G1 and g is a generator of G1G1. Next, *KGC* selects three secure hash function H1:{0,1}∗→G1,H2:G1→ {0,1}len,H3:{0,1}∗→Z∗qH1:{0,1} ∗→G1,H2:G1→{0,1}len,H3:{0,1} ∗→Zq∗. *KGC* publishes public parameters PK=(e,g,H1,H2,H3,h,si gg,θ,σ,fi=gsi,1≤i≤|A|)PK=(e,g,H1, H2,H3,h,sigg,θ,σ,fi=gsi,1≤i≤|A|) an d keeps master key $MK = (\alpha, \beta)$ secretly.

2. $\boldsymbol{KeyGen(MK, S) \rightarrow (Usk)}$**:** It is run by *KGC*. Before *DR* with attribute $\boldsymbol{S}$ share the data, he should get the private key to decrypt the cipher text of shared data. *DR* first sends key generation request $\boldsymbol{request_{KeyGen}(S)}$ to *KGC*. After receiving the request, *KGC* computes $\boldsymbol{Uid = H_3(S)}$ as the *DR* identity and computes $\boldsymbol{Upk = g^{Uid}}$ as *DR*'s public key. Next *KGC* selects t∈Zq,V,V′,Vx∈ G1,x∈St∈Zq,V,V′,Vx∈G1,x∈S and computes the private key ***Usk*** for *DR* as follows: $\boldsymbol{K = g^{\alpha}\sigma^{t}V, K' = g^{t}V', K_x = f_x^{t}V_{x}, x \in S}$. *KGC* randomly selects $\boldsymbol{b \in Z_q}$ and computes $\boldsymbol{Ub = Uid \cdot b}$ and sends $\{\boldsymbol{Ub}, \boldsymbol{g^b}\}$ to *CMS* for later integrity verification . Then, *KGC* sends $\{\boldsymbol{Uid}, \boldsymbol{Usk}\}$ to *DR* and $\{\boldsymbol{S}, \boldsymbol{Uid}, \boldsymbol{Upk}\}$ to *CSS*.
3. $\boldsymbol{IdReg(S, Uid) \rightarrow DRTable}$. It is run by *CSS*. To verify the validity of *DR* before transferring the shared data, *CSS* stores *DR*'s information including identity ***Uid***, attribute set $\boldsymbol{S}$ and public key ***Upk*** in a table named ***DRTable***. If identity of DR is valid, *CSS* transfers shared data to him. Otherwise, *CSS* refuses the download request of *DR*.



Ti=sigg(mi⊕H1(ti‖vi))Ti=sigg(mi⊕H1(ti ‖vi)) (1)

and sends $\{(m_i, T_i)_{,i \in [1, N]}\}$ to CSS.

1. $\boldsymbol{InterEnc(M) \rightarrow (C_i, E_i)}$. It is run by *CMS*. To decrease the computation burden of mobile devices at *DO* side, *CMS* helps to compute the intermediate encryption data. He selects $r_i \in Z_{q,i \in [1, l]}$ and computes Ci=σλif−riρ(i),Ei=griCi= σλifρ(i)−ri,Ei=gri. For later integrity verification, *CMS* stores the intermediate data C_i, , E_i locally.

Integrity verification phase

When *DR* wants to access shared data, he first sends integrity request to *CMS* for

verifying whether the data is intact. Then *CMS* generates integrity challenge ***ch*** and sends it to *CSS*. After getting the challenge, *CSS* computes data proof ***P*** and sends it to *CMS* to verify the integrity. This phase consists of the following three algorithms and Fig. 4 describes the data flow of the phase.

1. *ChalGen*(*Fid*) → (*ch*): Before downloading shared data, *DR* first sends request $request_{challen}(Fid, Uid)$ to *CMS* for integrity verification. After receiving the request, *CMS* verifies the validity of *DR*'s identity with equation $e(Upk, g^b) = e(g, g)^{Ub}$. If the equation does not hold, *CMS* rejects the request. Otherwise, *CMS* randomly selects *c* blocks and corresponding random numbers Ri∈Z∗qRi∈Zq∗. Then *CMS* sends challenge $ch = (i, R_i)_{,i \in [1,c]}$ to *CSS*.
2. *ProfGen*($T_i, F^{'}$) → (*P*): After receiving *ch* from *CMS*, *CSS* comp utes data proof and tag proof

DP=sigg(∑ci=1(mi⊕h(Ri)))DP=sigg(∑i=1c(mi⊕h(Ri)))
(2)

TP=∑ci=1Ti.TP=∑i=1cTi.
(3)

Then he sends proof $P = (DP, TP)$ to *CMS*.

3.*IntegityVer*(*P*) → (*true*, *false*).After getting the proof *P* , *CMS* computes L=∑ci=1sigg(h(Ri)⊕H1(ti‖vi))L=∑i=1csigg(h(Ri)⊕H1(ti‖vi)) and verifies whether the following equation holds.

DP=TP⊕LDP=TP⊕L
(4)

If the equation holds, it indicates that the data is intact and outputs *true*, Otherwise, *CMS* outputs *false*.

Conclusion

In this paper, we propose an efficient and secure data sharing scheme for mobile devices. The scheme guarantees security and authorized access of shared sensitive data. Furtherly, the scheme realizes efficient integrity verification before *DR* shares the data to avoid incorrect computation. Finally, the scheme achieves lightweight operations of mobile terminals on both *DO* and *DR* sides.

References

1. Farahat IS, Tolba AS (2018) A secure real-time internet of medical smart things (IOMST). Comput Electrical Eng 72:455–467
2. Rahmani AM, Gia TN, Negash KB (2018) Exploiting smart e-Health gateways at the edge of healthcare Internet-of-Things: A fog computing approach. Futur Gener Comput Syst 78:641–658
3. Zhang Y, Qiu M, Tsai C, Hassan M, Alamri A (2017) Health-CPS: healthcare cyber-physical system assisted by cloud and big data. IEEE Syst J 11:88–95
4. Ghazvini A, Shukur Z (2013) Security challenges and success factors of electronic healthcare system. Proc Technol 11:212–219
5. Guan Z, Lv Z, Du X et al (2019) Achieving data utility-privacy tradeoff in internet of medical things: a machine learning approach. Futur Gener Comput Syst 98:60–68
6. Elhoseny M, Abdelaziz A (2018) A hybrid model of internet of things

and cloud computing to manage big data in health services applications. Futur Gener Comput Syst 86:1383–1394
7. Han K, Li Q, Deng Z (2016) Security and efficiency data sharing scheme for cloud storage. Chaos Solitons Fractals 86:107–116
8. Gao F, Sunyaev A et al (2019) Context matters: a review of the determinant factors in the decision to adopt cloud computing in healthcare. Int J Inf Manag 48:120–138
9. Chang V (2017) Towards data analysis for weather cloud computing. Knowl Based Syst 127:29–45
10. Goyal V, Pandy O, Sahai A, Waters B (2006) Attribute-based encryption for fine-grained access control of encrypted data. In: Proceedings of the 13th ACM conference on Computer and communications security, p 89

Author Details:

Student details:

Name:Badiginchala Manohari

Mail:Manuh8226@gmail.com

Dr.Samuel George Institute of Engineering & Technology, Markapur, India

Guide details:

D.Kumar received B.Tech (CSIT) Degree from JNT University in 2006 and M.Tech (CSE) Degree from JNTUK Kakinada in 2011. He has 11 years of teaching experience. He joined as Assistant Professor in Dr.Samuel George Institute of Engineering & Technology, Markapur, India in 2006. Presently he is working as Associate Professor in CSE Dept. His Interested research areas are Image Processing and Computer Networks. He attended Various National Workshops and Conferences.